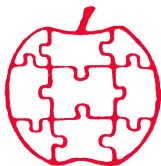


Apple



Assembly

Line

Volume 7 -- Issue 11

August, 1987

Using Double Lo-Res Graphics	2
Patch .TF to Assemble System Files	11
Viewing AppleWorks Word Processor Files.	15
Modify S-C Macro for Overlay Assembly.	20

New Chips for the IIgs

Apple is now shipping version 2 of the IIgs system ROM, and you are supposed to be able to get a free upgrade at your dealer. If your IIgs has the color fringe problem, they will also change a graphics chip. The new ROM fixes a lot of bugs in the various ROM-based tools, and adds some new features. One new feature I like is a new monitor command ("M") which installs the monitor as a desk accessory. Once done, the monitor is always available in a few keystrokes, and you can easily return to exactly the point from which you called it.

Naturally, the new ROM makes the "\$" command in the S-C Macro Assembler non-functional again. I will have to dig into the new code and see why, and fix it again. Unfortunately, as of the day I am typing this, I have not actually been able to upgrade MY IIgs. Mine is a prototype, and I have the "airplane" ROM board where Apple's single ROM chip is supposed to be. Maybe they will send me the appropriate upgrade soon, or maybe I'll be able to make a copy into 27256 EPROMs.

New Apple Accessories Coming Soon

You have read about Applied Engineering's PC Transporter in A+ magazine. But how about the ZIP CHIP from ZIP Technology? Or the GSX IIgs Accelerator from MDIdeas? Or the Memory Saver from Checkmate? ZIP CHIP is one of several new accelerators for pre-IIgs machines which are each a single chip that replace the 6502 or 65C02. The replacement contains the equivalent of a 4 MHz 65C02 with its own clock circuit and some super fast cache RAM; it is supposed to be inexpensive, compatible with everything, and make it all happen 3 or 4 times faster. GSX is a more traditional accelerator, fitting in any IIgs slot (1-7), and having a faster 65816 and RAM than is on the motherboard. It can run at 5.6 MHz and can include as much as 1 Mbyte of fast static RAM. Memory Saver turns an ordinary Apple RAM card or AE gsRAM card into a battery-backed up RAM, much like RamCharger does for RamFactor. All available soon, they say....

Using Double Lo-Res Graphics.....Bob Sander-Cederlof

That's right, low resolution, not high. Apple never talks about it, but the //e, //c, and IIgs all have a double low resolution mode, displaying either 40 or 48 lines of 80 pixels, in 16 colors. I know it is not in fashion, but lo-res still has some advantages over hi-res. For one, it is faster; for another, it takes less memory.

Subroutines to plot in lo-res graphics are included in the monitor firmware, and statements are included in both Integer BASIC and Applesoft. I have written some subroutines which provide all the same functions in double lo-res mode, and also added BOX and LINE subroutines. Since all //c and IIgs machines can execute the 65C02 extensions, and since all enhanced //e's also can, I decided to include some 65C02 instructions. They make the code a lot easier to write. I used the PHY, PLY, and BRA opcodes; I also used the INC opcode with no operand to increment the A-register. If you have an old unenhanced //e, you will have to re-write the code where these instructions occur.

Turning Double Lo-Res On and Off:

The same soft switch that changes normal hi-res into double hi-res does the trick for lo-res also. Reading or writing \$C05E will double the resolution, and \$C05F will put it back to normal. You also have to be in 80-column mode.

In 40-column mode with the "double" switch turned on you get a weird result: You still have only 40 pixels per line, and only four colors rather than 16. Furthermore, somehow which color is displayed for a particular pixel value depends on which column you are in! This is not a particularly useful mode, so I am not going to talk about it any more. Let's just stay in 80-column mode when we want to use double lo-res.

Applesoft's "GR" statement turns on the Lo-Res mode, clears the top 40 graphics lines, and sets the text window to the bottom four text lines. My subroutine "DLR", in lines 1700-1780, only turns on the double lo-res mode. I did not include clearing the graphics screen, or setting the text window. I included a separate subroutine "CLRTOP" for clearing the graphics screen, and "SETTOP" for setting the entire graphics screen to whatever the current color is (lines 2220-2250).

Lines 1640-1670 turn the double lo-res graphics back off, resetting to text mode. You could add a line here to clear the text screen and restore a full text window, like the Applesoft TEXT statement, if you wish.

Setting the Plotting Color:

The monitor firmware contains two entry points that my code uses: SETCOL (\$F864) and PLOT (\$F800). SETCOL uses the low-order four bits in the A-register to store a color value at COLOR (\$30). This value is formed by stuffing the 4-bit color value in both the left and right nybbles of COLOR.

S-C Macro Assembler Version 2.0.....DOS \$100, ProDOS \$100, both for \$120
Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....\$20
ProDOS Upgrade Kit for Version 2.0 DOS owners.....\$30
Source Code of S-C Macro 2.0 (DOS or ProDOS).....each, additional \$100
S-C DisAssembler (ProDOS only)...without source code \$30, with source \$50
RAK-Ware DISASM (DOS only).....without source code \$30, with source \$50
ProVIEW (ProDOS only).....\$20
Full Screen Editor for S-C Macro (with complete source code).....\$49
S-C Cross Reference Utility.....without source code \$20, with source \$50
S-C Word Processor (with complete source code).....\$50
DP18 and DPFP, including complete source and object code.....\$50
ES-CAPE (Extended S-C Applesoft Program Editor).....
Including Version 2.0 With Source Code.....\$50
ES-CAPE Version 2.0 and Source Code Update (for Registered Owners)....\$30
Bag of Tricks 2 (Quality Software).....(\$49.95) \$45 *
MacASM -- Macro Assembler for Macintosh (Mainstay).....(\$150.00) \$50 *
S-C Documentor (complete commented source code of Applesoft ROMs).....\$50
Cross Assemblers for owners of S-C Macro Assembler.....\$32.50 to \$50 each
(Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,
8051, 8085, 1802/4/5, PDP-11, G1650/70, others)
Beagle Bros. Applesoft Compiler (ProDOS only).....(\$74.95) \$65 *
AAL Quarterly Disks.....each \$15, or any four for \$45
Each disk contains the source code from three issues of AAL:
Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.
(All source code is formatted for S-C Macro Assembler. Other assemblers
require some effort to convert file type and edit directives.)
Diskettes (with hub rings)..... package of 25 for \$15 *
Vinyl disk pages, 6"x8.5", hold two disks each.....10 for \$6 *
Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each
(Cardboard folders designed to fit 6"x9" Envelopes.) or \$25 per 100 *
Envelopes for Diskette Mailers..... 6 cents each
65802 Microprocessor (Western Design Center)..... \$25 *
quikLoader EPROM System (SCRG).....(\$179) \$170 *
PROMGRAMMER (SCRG).....(\$149.50) \$140 *
Minuteman 250 Uninterruptible Power Supply.....(\$359) \$320 +
"Apple //gs Technical Reference", Fischer.....(\$19.95) \$19 *
"65816/65802 Assembly Language Programming", Fischer.....(\$21.95) \$20 *
"Programming the 65816", Eyes.....(\$22.95) \$21 *
"Apple //e Reference Manual", Apple Computer.....(\$24.95) \$23 *
"Apple //c Reference Manual", Apple Computer.....(\$24.95) \$23 *
"ProDOS-8 Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *
"ProDOS-16 Technical Reference Manual", Apple Computer.....(\$29.95) \$27 *
"Apple IIgs Firmware Reference", Apple Computer.....(\$24.95) \$23 *
"ProDOS Inside and Out", Doms & Weishaar.....(\$16.95) \$16 *
"DOSTalk Scrapbook", Weishaar & Kersey.....(\$14.95) \$14 *
"Beneath Apple ProDOS", Worth & Lechner.....(\$19.95) \$18 *
"Beneath Apple DOS", Worth & Lechner.....(\$19.95) \$18 *
"Inside the Apple //c", Little.....(\$19.95) \$18 *
"Inside the Apple //e", Little.....(\$19.95) \$18 *
"Understanding the Apple //e", Sather.....(\$24.95) \$23 *
"Understanding the Apple II", Sather.....(\$22.95) \$21 *
"Apple II/Iie Troubleshooting & Repair Guide", Brenner.....(\$19.95) \$18 *
"Apple II Circuit Description", Gayler.....(\$22.95) \$21 *
"Assem. Language for Applesoft Programmers", Finley & Myers.....(\$18.95) \$18 *
"Now That You Know Apple Assembly Language...", Gilder.....(\$19.95) \$18 *
"Enhancing Your Apple II, vol. 1", Lancaster.....(\$15.95) \$15 *
"Enhancing Your Apple II, vol. 2", Lancaster.....(\$17.95) \$17 *
"Assembly Cookbook for the Apple II/Iie", Lancaster.....(\$21.95) \$20 *
"Microcomputer Graphics", Myers.....(\$14.95) \$14 *
"Assembly Lines -- the Book", Wagner.....(\$19.95) \$18 *
"AppleVisions", Bishop & Grossberger.....(\$39.95) \$36 *

* On these items add \$2.00 for the first item and
\$.75 for each additional item for US shipping.

+ Inquire for shipping cost.

Foreign customers inquire for postage needed.

Texas residents please add 7.25 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***

*** (214) 324-2050 ***

*** Master Card, VISA, Discover and American Express ***

Plotting Single Points:

The monitor PLOT subroutine at \$F800 will plot a single point on the normal Lo-Res screen. It expects the Y-coordinate in the A-register, and the X-coordinate in the Y-register. I set up my DPLOT subroutine the same way, but the X-coordinate can be any value from 0 to 79 instead of just 0 to 39. Double Lo-Res works the same way 80-column text does: columns 0, 2, 4, ...78 are stored in AuxRAM, and columns 1, 3, 5, ...79 are stored in MainRAM. Since the softswitches 80COL and 80STORE are both in 80-mode, the PAGE2 switch controls whether text screen addresses reference AUX or MAIN memory.

My DPLOT subroutine is in lines 1800-1960. I look at the least significant bit of the X-coordinate to determine whether the pixel will be in MAIN or AUX memory. If it will be in AUX memory, then I flip to AUX memory before calling PLOT. I also divide the X-coordinate by two, so that PLOT will put it in the right position in the selected bank. After PLOT has painted the pixel, I make sure I am back in MAIN memory at line 1930. I made my DPLOT subroutine save and restore the A- and Y-registers, even though the monitor PLOT subroutine did not. This made some of the other subroutines cleaner.

Lines 1800-1820 are a special entry to DPLOT, and plot the point whose coordinates are stored in the X1 and Y1 variables.

Horizontal and Vertical Lines:

Both Integer BASIC and Applesoft include the HLINE and VLINE statements for drawing horizontal and vertical lines. These are faster to draw and easier to implement than the more general line drawing algorithm included in Hi-Res graphics. Lines 1980-2080 are my double Lo-Res HLINE and VLINE subroutines. Like the ones in the monitor firmware, the coordinates for the beginning of the line must be in the A- and Y-registers; the other end of the line is specified by the value in X2 for HLINE, or in Y2 for VLINE.

My version of HLINE is slower than it ought to be, because I used the monitor PLOT subroutine inside DPLOT. This causes GBASCALC to be called for every pixel, even though it does not need to be called after the first pixel. I could write a second version of DPLOT which calls PLOT1 in the firmware, bypassing the GBASCALC call, but I decide not to bother.

Drawing Boxes:

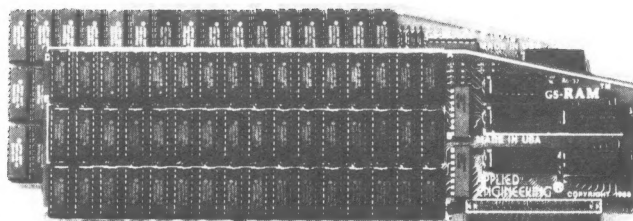
Lines 2100-2200 will draw a rectangle (box) of the currently selected color. The A- and Y-registers specify the lower-right corner of the rectangle; the variables X1 and Y1 specify the upper-left corner.

To make it easier to program BOX-calls, I defined three macros: FROM, BOX, and BOX.TO. ">FROM valueX,valueY" will assemble a call to SET.X1.Y1 in lines 3160-3200. This subroutine makes sure neither coordinate is beyond the screen and stores them in X1 and Y1. For example:

Insist on **gsRAM™** When You Buy Your **IIGs™**

*Expand the IIGs RAM and ROM with the gsRAM or gsRAM Plus
with ROM Pak Available now with 256K to 8 MEG!*

gsRAM Plus™



gsRAM™

Remember the 16K cards for the II+ and the 64K cards for the IIc? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

gsRAM and gsRAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only gsRAM and gsRAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in IIc-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

gsRAM for More AppleWorks Power

Only gsRAM and gsRAM Plus eliminate AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000, and only gsRAM and gsRAM Plus increase the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only gsRAM and gsRAM Plus offer a built-in printer buffer so you can continue using AppleWorks while your printer is printing. gsRAM and gsRAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can

even have Pinpoint or Macroworks and your favorite spelling checker in RAM for instant response. gsRAM and gsRAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIGs into a Giant

Simply plug gsRAM into the IIGs memory expansion slot and you've got up to 8 megabytes of RAM at your fingertips—all of it instantly and automatically recognized by the IIGs. gsRAM is compatible with all IIGs software, including AppleWorks, as well as BASIC®, ProDOS, DOS 3.3, PASCAL®, "C" and CP/M®.

Grow by Bytes or Megabytes

We offer gsRAM in two configurations so you can increase your memory 256K at a time (gsRAM) or a megabyte at a time (gsRAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded gsRAM operates at only 375 ma, and gsRAM Plus at only 270 ma (even with 6 megabytes on board!)

gsRAM—For Normal Memory Requirements

gsRAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a gsRAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to gsRAM Plus for a small charge.



Steve Wozniak, the creator
of Apple Computer

*In quality,
performance,
compatibility,
expandability and
support, Applied
Engineering's gsRAM
and gsRAM Plus are
number one.*

With an optional piggyback card, you can expand gsRAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

gsRAM Plus—for Growing by Leaps and Bounds

gsRAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a gsRAM Plus with less memory and easily expand it in the future.

gsRAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both gsRAM and gsRAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your gsRAM or gsRAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIGs's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We invented the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure gsRAM and gsRAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

gsRAM with 256K	\$169
gsRAM with 512K	\$219
gsRAM with 1 MEG	\$299
gsRAM with 1.5 MEG	\$379
gsRAM with 2-6 MEG	CALL
gsRAM PLUS with 1 MEG	\$459
gsRAM PLUS with 2 MEG	\$759
gsRAM PLUS with 3-8 MEG	CALL

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6% sales tax. Add \$10.00 outside USA.

AE APPLIED ENGINEERING™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 598, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research.

```

>FROM 3,10          (the macro call)
LDA #10             (the generated lines)
LDY #3              "
JSR SET.X1.Y1       "

```

```

>BOX.TO 9,13        (assumes X1,Y1 already set)
LDA #13
LDY #9
JSR BOX

```

>BOX 3,10,9,13 will generate a >FROM and a >BOX.TO macro call, which will then be expanded.

Drawing Diagonal Lines:

Or, more properly, "Drawing Lines from Any Point to Any Other Point". In Applesoft Hi-Res graphics you can use "HPLOT x,y TO x2,y2". I wrote a similar subroutine in lines 2300 and following. This subroutine assumes you have already stored the coordinates of the beginning of the line in X1 and Y1, and that the coordinates of the end of the line are in the A- and Y-registers. (Just like BOX did.)

To make it easy to write program which draw lines, I defined some more macros. Use >FROM to set up X1 and Y1, just as with BOX. Use >LINE.TO to draw a line from that point to another. Use >LINE if you want to specify both ends at once. After a line has been drawn, X1 and Y1 will contain the coordinates of the end of that line; this means that you can draw a continuous path around the screen by using a string of >LINE.TO calls.

The algorithm for drawing a straight line from one point to another is probably not familiar to all. I believe it was developed many years ago (mid 50's?) for the early XY-plotters. My implementation may be a little hasty, because I seem to remember there is a simplification that involves shuffling variables around to fold two cases into one, but it does work, is quick, and does not require any multiplication or division.

To be sure I had it right, I first wrote a lo-res line-drawing program in Applesoft. Here is the listing of that program to help you understand the assembly language in my LINE routine.

```

100 GR : COLOR= 15
110 INPUT X1,Y1,X2,Y2
120 GOSUB 1000
130 GOTO 110
200 GR
210 COLOR= INT ( RND (1) * 4) * 5
220 X1 = INT ( RND (1) * 40)
230 Y1 = INT ( RND (1) * 40)
240 X2 = INT ( RND (1) * 40)
250 Y2 = INT ( RND (1) * 40)
260 GOSUB 1000
270 IF PEEK ( - 16384) < 128 THEN 210
280 POKE - 16368,0
290 END
1000 REM PLOT X1,Y1 TO X2,Y2
1010 REM IN LO-RES WITH CURRENT COLOR
1020 DX = ABS (X2 - X1):DY = ABS (Y2 - Y1)
1030 SX = SGN (X2 - X1):SY = SGN (Y2 - Y1)
1040 X = X1:Y = Y1
1050 IF DX < DY THEN 1200
1060 D = DX - DY
1070 PLOT X,Y

```

```

1080 D = D - DY:X = X + SX: IF D < 1 THEN Y = Y + SY:D = D + DX
1090 IF X < > X2 THEN 1070
1100 PLOT X2,Y2
1110 RETURN
1200 D = DY - DX
1210 PLOT X,Y
1220 D = D - DX:Y = Y + SY: IF D < 1 THEN X = X + SX:D = D + DY
1230 IF Y < > Y2 THEN 1210
1240 PLOT X2,Y2
1250 RETURN

```

```

1010 *SAVE S.DOUBLE LO-RES
1020 .OP 65C02
1030 *-----
2C- 1040 X2 .EQ $2C
2D- 1050 Y2 .EQ $2D
30- 1060 COLOR .EQ $30
1070 *-----
F800- 1080 PLOT .EQ $F800
F864- 1090 SETCOL .EQ $F864
1100 *-----
1110 .MA FROM
1120 LDA #12 Y1
1130 LDY #11 X1
1140 JSR SET.X1.Y1
1150 .EM
1160 *-----
1170 .MA LINE.TO
1180 LDA #12 Y2
1190 LDY #11 X2
1200 JSR LINE
1210 .EM
1220 *-----
1230 .MA LINE
1240 >FROM 11,12 X1,Y1
1250 >LINE.TO 13,14 X2,Y2
1260 .EM
1270 *-----
1280 .MA BOX.TO
1290 LDA #12 Y2
1300 LDY #11 X2
1310 JSR BOX
1320 .EM
1330 *-----
1340 .MA BOX
1350 >FROM 11,12 X1,Y1
1360 >BOX.TO 13,14 X2,Y2
1370 .EM
1380 *-----
1390 T
0800- 20 8A 08 1400 JSR DLR Turn on Double Lo-Res
0803- 20 DF 08 1410 JSR CLRTOP Clear screen
0806- A9 0F 1420 LDA #15 Set Color = White
0808- 20 64 F8 1430 JSR SETCOL
1440 *---DRAW DIAGONAL LINES-----
080B- 1450 >LINE 0,0,39,39
0819- 1460 >LINE 0,39,39,0
0827- 1470 >LINE 79,39,40,0
0835- 1480 >LINE 79,0,40,39
1490 *---BOX 35,15 TO 44,24-----
0843- 1500 >BOX 35,15,44,24
1510 *---BORDER AROUND SCREEN-----
0851- 1520 >LINE 0,0,79,0
085F- 1530 >LINE.TO 79,39
0866- 1540 >LINE.TO 0,39
086D- 1550 >LINE.TO 0,0
1560 *-----
0874- AD 00 C0 1570 .2 LDA $C000
0877- 10 FB 1580 BPL .2
0879- 8D 10 C0 1590 STA $C010
1600 *-----
087C- 20 80 08 1610 JSR TEXT
087F- 60 1620 RTS
1630 *-----
1640 TEXT
0880- 8D 51 C0 1650 STA $C051 TEXT
0883- 8D 5F C0 1660 STA $C05F SINGLE
0886- 8D 54 C0 1670 STA $C054
0889- 60 1680 RTS

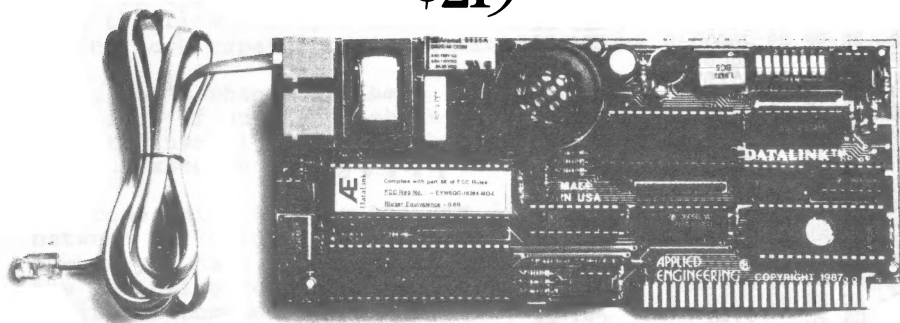
```

```

1690 *-----
1700 DLR
088A- A9 99 1710 LDA #99
088C- 20 00 C3 1720 JSR $C300 TURN ON 80-COLUMN MODE
088F- 8D 50 C0 1730 STA $C050 GRAPHICS
0892- 8D 53 C0 1740 STA $C053 MIXED TEXT/GRAPHICS
0895- 8D 54 C0 1750 STA $C054 START IN MAIN MEM
0898- 8D 56 C0 1760 STA $C056 LO-RES
089B- 8D 5E C0 1770 STA $C05E DOUBLE
089E- 60 1780 RTS
1790 *-----
1800 DPLOT.X1.Y1
089F- AD BF 09 1810 LDA Y1
08A2- AC BE 09 1820 LDY X1
08A5- 5A 1830 DPLOT PHY
08A6- 48 1840 PHA SAVE Y-COORD
08A7- 98 1850 TYA X-COORD
08A8- 4A 1860 LSR
08A9- A8 1870 TAY X/2
08AA- 68 1880 PLA
08AB- 48 1890 PHA
08AC- B0 03 1900 BCS .1
08AE- 8D 55 C0 1910 STA $C055
08B1- 20 00 F8 1920 .1 JSR PLOT
08B4- 8D 54 C0 1930 STA $C054
08B7- 68 1940 PLA
08B8- 7A 1950 PLY
08B9- 60 1960 RTS
1970 *-----
08BA- C8 1980 HLIN1 INY
08BB- 20 A5 08 1990 HLIN JSR DPLOT
08BE- C4 2C 2000 CPY X2
08C0- 90 F8 2010 BCC HLIN1
08C2- 60 2020 RTS
2030 *-----
08C3- 69 01 2040 VLIN1 ADC #1
08C5- 20 A5 08 2050 VLIN JSR DPLOT
08C8- C5 2D 2060 CMP Y2
08CA- 90 F7 2070 BCC VLIN1
08CC- 60 2080 RTS
2090 *-----
2100 * BOX X2 IN Y, Y2 IN A
2110 *-----
08CD- 20 B6 09 2120 BOX JSR SET.X2.Y2
08D0- AD BF 09 2130 LDA Y1
08D3- AC BE 09 2140 .1 LDY X1
08D6- 20 BB 08 2150 JSR HLIN
08D9- C5 2D 2160 CMP Y2
08DB- 1A 2170 INC
08DC- 90 F5 2180 BCC .1
08DE- 60 2190 RTS
2200 *-----
08DF- A9 00 2210 CLRTOP LDA #0
08E1- 85 30 2220 STA COLOR
08E3- 2230 SETTOP >BOX 0,0,79,39
08F1- 60 2240 RTS
2250 *-----
2260 * LINE FROM X1,Y1 TO X2,Y2
2270 * X2 IN Y, Y2 IN A
2280 *-----
08F2- 20 B6 09 2290 LINE JSR SET.X2.Y2
08F5- A2 01 2300 LDX #1 First Y's, then X's
08F7- A0 00 2310 .1 LDY #0 START WITH SGN( )=0
08F9- 38 2320 SEC
08FA- B5 2C 2330 LDA X2,X X2 or Y2
08FC- FD BE 09 2340 SBC X1,X X1 or Y1
08FF- F0 08 2350 BEQ .3 VERTICAL or HORIZONTAL
0901- 10 05 2360 BPL .2 Positive DELTA X or DELTA Y
0903- 88 2370 DEY Negative DELTA X or DELTA Y
0904- 49 FF 2380 EOR #$FF 2's Complement
0906- 1A 2390 INC
0907- 88 2400 DEY (overcome following INY)
0908- C8 2410 .2 INY SGN( ) = -1, 0, +1
0909- 9D C2 09 2420 .3 STA DX,X DX or DY
090C- 98 2430 TYA
090D- 9D C0 09 2440 STA SX,X SX or SY
0910- CA 2450 DEX
0911- 10 E4 2460 BPL .1
2470 *-----

```


Global communications. Down-to-earth price. \$219



Introducing the Datalink™ modem.
Get instant access to networks and databases
— plus network memberships and discounts worth \$177.95!

Now you can tap into a world of information — without draining your resources. Our Datalink™ modem lets your Apple IIGS, IIe or II+ communicate with other computers, download free software from networks and bulletin boards, access database services and more. For a remarkably low price.

The world at your fingertips.

With the Datalink modem, you'll be able to draw information from thousands of databases. Send and receive electronic mail — even overseas. Join clubs and bulletin boards to exchange software or solve computing problems with other Apple users. Download free public domain software. And share files with personal computers and mainframes.

There's nothing missing on Datalink.

Unlike the Smartmodem®, Datalink comes with its own easy-to-use communications software in ROM and on disk that supports macros, file transfers, on-line time display, data capture and datascopes mode. The software also allows you to store hundreds of phone numbers for auto dialing and log on.

Because Datalink is 100% Super Serial Card compatible, you can also run virtually all other communication software, including Point-to-Point, Apple Access and ASCII Express, to name a few.

The compact design allows Datalink to fit in any slot — including slots 1 and 2 of the IIGS with a cooling fan installed. Datalink operates at 1200 or 300 baud. Built-in diagnostics check for accuracy of data transmission, and Datalink lets you track the progress of calls either electronically or via an on-board speaker.

Introductory offers from popular networks included.

When you purchase Datalink, you'll get a fee-waived membership to The Source worth \$49.95. \$60.00 worth of free on-line time from News Net. A free \$50.00 subscription to the Official Airline Guide. A free subscription to the GENIE network worth \$18.00! And information on MCI Mail.

Order today!

To order or for more information, see your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 6 1/2% sales tax. Add \$10.00 outside U.S.A.

	Applied Engineering Datalink	Hayes® Smartmodem 1200A
Price	\$219	\$439
Max. transmission rate	1200 baud	1200 baud
Warranty period	5 years at no charge	2 years or 4 years for \$75
Software included	YES	NO
Hayes AT command set	YES	YES
Help screens	YES	NO
On-board telephone jacks	YES	NO
Fits any slot (even with fan)	YES	NO

Smartmodem is a registered trademark of D.C. Hayes Corporation

AE Applied Engineering
The Apple enhancement experts.

Box 798, Carrollton, TX 75006
(214) 241-6060

0913-	38		2480	SEC		
0914-	AD C3	09	2490	LDA DY		
0917-	ED C2	09	2500	SBC DX	DY-DX	
091A-	10 43		2510	BPL .8	...DX < DY	
091C-	49 FF		2520	EOR ##FF	2'S COMPLEMENT	
091E-	1A		2530	INC		
091F-	8D C4	09	2540	STA D		
0922-	20 9F	08	2550	.5 JSR DPLLOT.X1.Y1		
0925-	18		2560	CLC		
0926-	AD BE	09	2570	LDA X1		
0929-	6D C0	09	2580	ADC SX		
092C-	8D BE	09	2590	STA X1		
092F-	38		2600	SEC		
0930-	AD C4	09	2610	LDA D		
0933-	ED C3	09	2620	SBC DY		
0936-	8D C4	09	2630	STA D		
0939-	F0 02		2640	BEQ .6		
093B-	10 13		2650	BPL .7		
093D-	18		2660	.6 CLC	Y = Y + SY	
093E-	AD BF	09	2670	LDA Y1		
0941-	6D C1	09	2680	ADC SY		
0944-	8D BF	09	2690	STA Y1		
0947-	AD C4	09	2700	LDA D	D = D + DX	
094A-	6D C2	09	2710	ADC DX		
094D-	8D C4	09	2720	STA D		
0950-	AC BE	09	2730	.7 LDY X1		
0953-	C4 2C		2740	CPY X2		
0955-	D0 CB		2750	BNE .5		
0957-	A5 2D		2760	LDA Y2		
0959-	8D BF	09	2770	STA Y1		
095C-	4C A5	08	2780	JMP DPLLOT		
2790				*-----		
095F-	8D C4	09	2800	.8 STA D		
0962-	20 9F	08	2810	.9 JSR DPLLOT.X1.Y1		
0965-	18		2820	CLC		
0966-	AD BF	09	2830	LDA Y1	Y = Y + SY	
0969-	6D C1	09	2840	ADC SY		
096C-	8D BF	09	2850	STA Y1		
096F-	38		2860	SEC	D = D - DX	
0970-	AD C4	09	2870	LDA D		
0973-	ED C2	09	2880	SBC DX		
0976-	8D C4	09	2890	STA D		
0979-	F0 02		2900	BEQ .10	...D = 0	
097B-	10 13		2910	BPL .11	...D > 0	
097D-	18		2920	.10 CLC	X = X + SX	
097E-	AD BE	09	2930	LDA X1		
0981-	6D C0	09	2940	ADC SX		
0984-	8D BE	09	2950	STA X1		
0987-	AD C4	09	2960	LDA D	D = D + DY	
098A-	6D C3	09	2970	ADC DY		
098D-	8D C4	09	2980	STA D		
0990-	AD BF	09	2990	.11 LDA Y1		
0993-	C5 2D		3000	CMP Y2		
0995-	D0 CB		3010	BNE .9		
0997-	A4 2C		3020	LDY X2		
0999-	8C BE	09	3030	STY X1		
099C-	4C A5	08	3040	JMP DPLLOT		
3050				*-----		
3060				LIMIT.XY		
099F-	C9 28		3070	CMP #40		
09A1-	90 02		3080	BCC .1		
09A3-	A9 27		3090	LDA #39		
09A5-	C0 50		3100	.1 CPY #80		
09A7-	90 02		3110	BCC .2		
09A9-	A0 4F		3120	LDY #79		
09AB-	60		3130	.2 RTS		
3140				*-----		
3150				SET.X1.Y1		
09AC-	20 9F	09	3160	JSR LIMIT.XY		
09AF-	8C BE	09	3170	STY X1		
09B2-	8D BF	09	3180	STA Y1		
09B5-	60		3190	RTS		
3200				*-----		
09B6-	20 9F	09	3210	SET.X2.Y2	09BE-	3270 X1 .BS 1
09B9-	84 2C		3220	JSR LIMIT.XY	09BF-	3280 Y1 .BS 1
09BB-	85 2D		3230	STY X2	09C0-	3290 SX .BS 1
09BD-	60		3240	STA Y2	09C1-	3300 SY .BS 1
3250				RTS	09C2-	3310 DX .BS 1
3260				*-----	09C3-	3320 DY .BS 1
					09C4-	3330 D .BS 1

Patch .TF to Assemble System Files.....Bob Sander-Cederlof

The ProDOS version of S-C Macro Assembler can put your object code directly on a target file, by using the .TF directive to specify the file name. The way it is designed, you have no control over the file type: it always comes out type BIN, which is type \$06. Sometimes it would be nice to be able to have the file end up a different file type, such as SYS (\$FF), but no such capability is available in the S-C Macro Assembler.

Well, let's change it then! At least temporarily, whenever we need it. If such a patch is possible, we could make an EXEC file to install the patch whenever we need it, and simply re-load when we want to go back to normal.

The code which handles the .TF directive is located somewhere between \$8F70 and \$9000, depending on the particular serial number you have. The way to locate it in yours is to look for the following code somewhere between \$8F70 and \$9000. (Use the monitor L command to do the looking.)

A9 00	LDA #0	
8D C8 BE	STA \$BEC8	
8D C9 BE	STA \$BEC9	
8D CA BE	STA \$BECA	
AD D0 BE	LDA \$BED0	
85 98	STA \$98	
8D C7 BE	STA \$BEC7	
A9 CE	LDA #\$CE	Set Mark = 000000
20 70 BE	JSR \$BE70	
B0 ..	BCS ...	
A9 D0	LDA #\$D0	Set End-of-File = 000000
20 70 BE	JSR \$BE70	
B0 ..	BCS ...	
A9 0A	LDA #\$0A	10 Parameters
8D B4 BE	STA \$BEB4	
A9 C4	LDA #\$C4	Get File Info
20 70 BE	JSR \$BE70	
B0 ..	BCS ...	
A9 07	LDA #\$07	7 Parameters
8D B4 BE	STA \$BEB4	
** A9 06	LDA #\$06	File Type = \$06 (BIN)
8D B8 BE	STA \$BEB8	
A5 BC	LDA \$BC	Lo-byte of Origin
8D B9 BE	STA \$BEB9	into AuxType
A5 BD	LDA \$BD	Hi-byte of Origin
8D BA BE	STA \$BEBA	into AuxType+1
A9 C3	LDA #\$C3	Set File Info
20 70 BE	JSR \$BE70	
B0 ..	BCSerror

It is the starred line "LDA #\$06" which we would need to change to get a different file type. By changing it to "LDA #\$FF" we would get type SYS files for .TF directives. We might also want to change the lines which store into the AuxType field. AuxType holds the Load Address for BIN files, but SYS files always load at \$2000 so AuxType is not used.

Once you find the code above, and determine the address of the file type value (the 06 in the LDA #\$06 instruction), you can change it with a simple monitor command. Or, you could get fancy and write a program to do the searching for you and make the patch.

And I wrote just such a program! The following code will search for the patch location, and if it finds \$06 in the file type address it will change it to \$FF. If it finds anything else there, it will change it back to \$06. And it will print out a message telling which one it did.

```

1000 *SAVE S.PATCH.TF
1010 *-----
00- 1020 PNTR .EQ $00,01
1030 *-----
FD8D- 1040 MON.COUT .EQ $FD8D
FD8E- 1050 MON.CROUT .EQ $FD8E
1060 *-----
1070 T
0800- A9 00 1080 LDA #$8E00 START LOOKING AT $8E00
0802- 85 00 1090 STA PNTR
0804- A9 8E 1100 LDA /$8E00
0806- 85 01 1110 STA PNTR+1
1120 *-----
0808- A0 00 1130 .1 LDY #0
080A- B1 00 1140 .2 LDA (PNTR),Y
080C- D9 41 08 1150 CMP KEY,Y
080F- F0 11 1160 BEQ .3 ...MATCHES SO FAR
0811- E6 00 1170 INC PNTR SLIDE WINDOW DOWN ONE
0813- D0 F3 1180 BNE .1
0815- E6 01 1190 INC PNTR+1
0817- A5 01 1200 LDA PNTR+1 STOP BEFORE IT GOES TOO FAR
0819- C9 91 1210 CMP /$9100 $90FF IS PLENTY FAR ENOUGH
081B- 90 EB 1220 BCC .1 ...NOT THERE YET
081D- A0 00 1230 LDY #Q.NOT.FOUND
081F- 4C 4D 08 1240 JMP MSG.OUT
1250 *-----
0822- C8 1260 .3 INY NEXT BYTE IN KEY AND WINDOW
0823- C0 06 1270 CPY #KEY.LEN
0825- 90 E3 1280 BCC .2 ...MORE IN THE KEY
1290 *-----
0827- A0 06 1300 LDY #PLOC-KEY
0829- B1 00 1310 LDA (PNTR),Y GET CURRENT FILETYPE
082B- C9 06 1320 CMP #$06
082D- F0 09 1330 BEQ .4 ...NOW "BIN", MAKE IT "SYS"
082F- A9 06 1340 LDA #$06 ...NOT OTHER, MAKE IT "BIN"
0831- 91 00 1350 STA (PNTR),Y
0833- A0 0D 1360 LDY #Q.MADE.BIN
0835- 4C 4D 08 1370 JMP MSG.OUT
1380 *-----
0838- A9 FF 1390 .4 LDA #$FF MAKE IT TYPE "SYS"
083A- 91 00 1400 STA (PNTR),Y
083C- A0 26 1410 LDY #Q.MADE.SYS
083E- 4C 4D 08 1420 JMP MSG.OUT
1430 *-----
0841- A9 07 1440 KEY LDA #$07
0843- 8D B4 BE 1450 STA $BEB4
0846- A9 1460 .HS A9 "LDA" OF "LDA #$06"
0847- 1470 PLOC .EQ *
06- 1480 KEY.LEN .EQ *-KEY
1490 *-----
0847- 09 80 1500 MSG1 ORA #$80
0849- 20 ED FD 1510 JSR MON.COUT
084C- C8 1520 INY
1530 MSG.OUT
084D- B9 58 08 1540 LDA QTS,Y
0850- 10 F5 1550 BPL MSG1
0852- 20 ED FD 1560 JSR MON.COUT
0855- 4C 8E FD 1570 JMP MON.CROUT
1580 *-----

```

```

1590 QTS
1600 Q.NOT.FOUND .EQ #-QTS

00-
0858- 4B 45 59
085B- 20 4E 4F
085E- 54 20 46
0861- 4F 55 4E
0864- C4
1610 .AT /KEY NOT FOUND/
1620 Q.MADE.BIN .EQ #-QTS

00-
0865- 50 41 54
0868- 43 48 45
086B- 44 20 54
086E- 4F 20 46
0871- 49 4C 45
0874- 54 59 50
0877- 45 20 22
087A- 42 49 4E
087D- A2
1630 .AT /PATCHED TO FILETYPE "BIN"/
1640 Q.MADE.SYS .EQ #-QTS

26-
087E- 50 41 54
0881- 43 48 45
0884- 44 20 54
0887- 4F 20 46
088A- 49 4C 45
088D- 54 59 50
0890- 45 20 22
0893- 53 59 53
0896- A2
1650 .AT /PATCHED TO FILETYPE "SYS"/
1660 #-----

```

EnterSoft

Basic-like Macros for the S-C Macro Assembler. These macros are Integer Basic Macros which can perform most of the standard Basic commands using 8/16/32/64 bit manipulation. Included are a set of graphic commands to perform block graphics and other Hi-Res Animation. Available in either DOS 3.3 or ProDos Format. Either one for \$30.00 or BOTH for \$50.00.

A Shape Table Program:

A shape table program unlike any other program you have ever run. This one is completely set up in a logical manner. All disk operations are under one section. All editing functions are contained in one place. All Hi-Res functions are in one place. The entire source code is included to allow you to modify it as you need to! The DOS 3.3 Version is only \$20.00 - AND SO IS THE PRODOS VERSION!!!! Both Versions for only \$30.00. ORDER YOURS TODAY!!!!

To Order:

Send check or Money Order To:

Mark Manning
c/o Simulacron 1/Baggy Games
P.O. Box 591894
Houston, Tx 77259-1894

Upgrades for EnterSoft from DOS 3.3 to ProDos is \$20.00. Get yours today!

Thanks Everyone For Your
Comments and Support!



SPECIAL !!! EXPANDED RAM/ROM BOARD: \$39.00

Similar to our \$30 RAM/ROM dev board described below. Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

IMPROVED !!! II IN A MAC (ver 2.0): \$75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

SCREEN.GEN: \$35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIC. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

MIDI-MAGIC for Apple II/c: \$49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only \$99.00).

FONT DOWNLOADER & EDITOR: \$39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, II/c. Specify printer: Apple Imagewriter, Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

*** FONT LIBRARY DISKETTE #1: \$19.00** contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e : \$30.00 (\$50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: \$39.00 (\$59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Grafrax-80, MX-100, MX-80/100 with Grafraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), Okidata 82A/83A with Okigraph & Okidata 92/93.

'MIRROR' ROM: \$25.00 (\$45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: \$30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into \$Cn00-CnFF & \$C800-CFFF.

C-PRINT For The APPLE II/c: \$69.00

Connect standard parallel printers to an Apple II/c serial port. Separate P/S included. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a \$3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange NJ 07052 (201) 325-1885



I don't happen to be an owner of Appleworks. This may be the tiniest minority of which I am a member. However I do see a lot of type AWP files on disks you send me. Problem, how do I really "see" them? I need a way to list them to the screen so that they look the way you wrote them: a WISIWYS lister (What I See Is What You Saw). It strikes me now that people who are trying to "hear" the files using a Speech Synthesizer might also need some special lister.

I have been doing it by BLOADing the file I need to see, specifying a starting address and file type, like this:

```
:BLOAD MSG.TO.SC,A$1000,TAWP
```

That leading colon is the prompt from the S-C Macro Assembler, which is usually where I am when I do the BLOAD. Then a hex-dump with ASCII on the right, lets me read it. But it is an uncomfortable way to read.

I looked up the format of the Appleworks Word Processor files, and decided to write a quick-and-dirty lister. The first 300 bytes of a type AWP file are of no interest to my lister. This header contains the tab stops and a few flags, and I don't know what else. My documentation is from February 1984, and most of the bytes were undefined at that time. I presume they are better defined now that over three years of improvements have been made to Appleworks, but I don't have that documentation.

If I BLOAD at address \$1000, then 300 bytes in starts at \$112C. From there to the end of file there are basically two types of data records: commands and text. Commands are two-byte records, and text lines consist of a four-byte header and a variable amount of text.

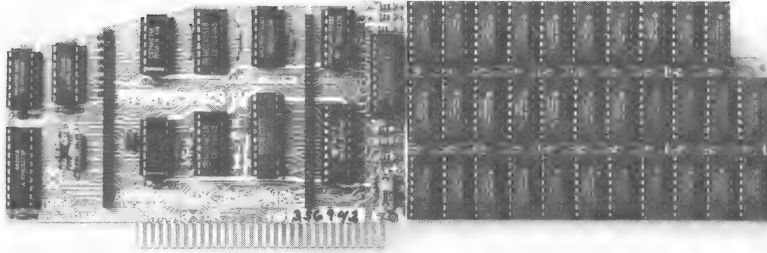
You distinguish whether a record is a command or text line by looking at the second byte. If it is 00, you have a text line. If it is \$D0-\$FF, you have a command. If it is anything else, I don't know what you have. Maybe garbage?

If the second byte of a record is \$D0 or larger, indicating a command, then the first byte of the record may contain more information for the command. The most frequent command is \$D0, which means a carriage return command. In my lister I simply output a carriage return when I see this command. Another popular command is \$FF, which means end-of-file. Obviously, when my lister finds this code it is time to end the listing! All the other codes I chose to simply list in hexadecimal. To distinguish them from text they list on separate lines of the form ".\$xx.\$yy", where xx is the command code (second byte) and yy is the operand value (first byte).

Text lines have the second byte = \$00. The first byte of these records contains a byte count, but it is somewhat redundant: it is always two larger than another byte count found in bits 6-0 of the fourth byte of the record. I chose to use this second byte count, which is the actual number of bytes of ASCII

RamWorks® III

Patented Performance from the Recognized Leader



The Best Selling, Most Compatible, Most Recommended, Most Expandable Card Available.

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 265K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 22,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 22,000. *Only* RamWorks expands the maximum clipboard limit from 250 to 2,042 lines. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your

printer to stop before returning to AppleWorks. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, *nothing* comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, Howard-Soft, FlashCalc, Pro-FILER, Managing Your Money, SuperCalc 3a and VIP Professional to name a few (and most hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601018) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all Apple IIe's enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and

RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 MEG is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 1 MEG, or 2 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector.

It Even Corrects Mistakes.

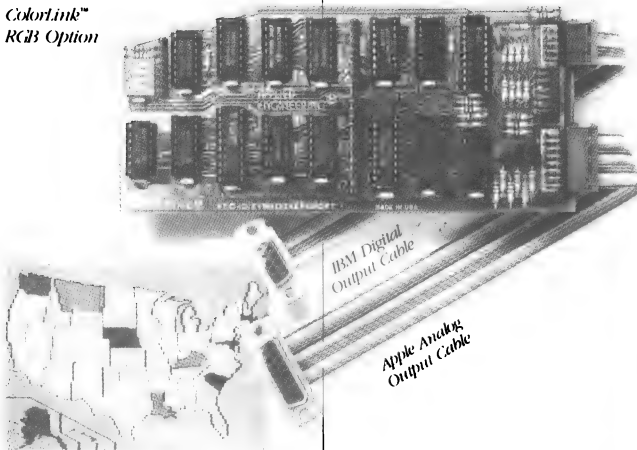
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any

composite color monitor). For only \$129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both the new Apple standard analog and the IBM standard digital RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 interference and remember you can order the RGB option with your RamWorks or add it on at a later date.

ColorLink™ RGB Option



True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 MEG of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 while maintaining full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." inCider

magazine said "RamWorks is the most powerful auxiliary slot memory card available for your IIe, and I rate it four stars. For my money, Applied Engineering's RamWorks is king of the hill."



*Steve Wozniak, the creator
of Apple Computer*

*"I wanted a
memory card for
my Apple that was
fast, easy to use,
and very compat-
ible, so I bought
RamWorks."*

Apple experts everywhere are impressed by RamWorks' expandability, versatility, ease of use, and the sheer power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PCs and ATs look like slowpokes.

- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 301"K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- 100% compatible with all standard IIe software
- RamDrive™ and ProDrive™ the ultimate emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible RGB option featuring ultra high resolution color graphics and multiple text colors with cables for both the new Apple and standard IBM type monitors
- Self diagnostics software included
- Lowest power consumption (U.S. Patent #4601018)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Wozniak and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	\$179
RamWorks III with 256K	\$199
RamWorks III with 512K	\$249
RamWorks III with 1 MEG	\$329
RamWorks III with 1.5 MEG	\$489
RamWorks III with 3.0 MEG	\$1099
Ram Expander+ 1 MEG	\$499
Ram Expander+ 2 MEG	\$799
65C816 16 Bit Card	\$159
ColorLink RGB Option	\$129
Optional Software:	
Pinpoint with RAM	
Enhancement Software	\$79
VIP Professional	\$219

Quality and Support of the Industry Leader.

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All.

- 15 day money back guarantee
- 5 year hassle free warranty
- Built-in super sharp 80 column display (U.S. Patent #4601018)
- Expandable to 1 MEG on main card
- Expansion connector for possible future expansion to 16 MEG

RamWorks III. The industry standard for memory expansion of the Apple IIe. ORDER YOU'R RamWorks III TODAY. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering. MasterCard, Visa and C.O.D. welcome. Texas residents add 6 1/2% sales tax. Add \$10.00 if outside U.S.A.

AE APPLIED ENGINEERING™
The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75086

and control codes which follow the four-byte header. The third byte of the text line header is supposed to be a leading blank count for the line, but I haven't found anything but zero there in the files I have viewed. Anyway, I chose to ignore this byte in my lister. The sign bit of the fourth byte indicates whether or not a carriage return should be tacked onto the end of this line.

In a paragraph, separate screen lines will be separate text line records. All except the last line will have bit 7 of the fourth header byte = zero, while the last line of the paragraph will have that bit = one. In my lister I decided to allow two ways of viewing the file. If you assemble with WRAP in line 1020 = 0, a file will list with carriage returns at the end of every text line record. This makes it look like it did inside the Appleworks Word Processor. If you change line 1020 to WRAP .EQ 1, no carriage return will be listed until the end of the paragraph.

Within text line records you have both ASCII characters (\$20-\$7F) and control codes (\$00-\$1F). The control codes are NOT the same as ASCII control characters. They have special meaning within Appleworks. The documentation I have defines codes \$01 through \$0B as follows:

Begin	End	
01	02	Boldface
03	04	Superscript
05	06	Subscript
07	08	Underline
09	--	Print Page Number
0A	--	Enter Keyboard
0B	--	Sticky Space

I converted \$0B (sticky space) to \$20 (ASCII space). A sticky space is supposed to print as a space, but not allow splitting at that point at the end of a line. All of the other codes I chose to simply ignore. You could obviously do something special with them. Either act intelligently and do on the screen or printer what they are supposed to do, or simply print them out as "<xx>" or the like.

You could add code to this lister to make it into a super-duper program. I thought about adding the front end to my ProDOS QUIT code program (from July, 1986 AAL). I would change the file type selection from that front end from SYS and BIN to AWP. Then I could easily find any AWP file on any mounted volume using the arrow keys, and BLOAD it with the RETURN key.

The next nice addition would be an output option section that would allow sending the output to a printer, speech synthesizer, or a TXT file.

If you do add either or both of these, or some other nice features, how about sending them here?

1000 *SAVE LIST.AWP

```

00-      1020 *-----
1030 WRAP .EQ 0 for CR at end of each line
1040      1 for CR only at end of paragraph
00-      1050 *-----
1060 P .EQ $00,01
1070 *-----
FD8E- 1080 CROUT .EQ $FD8E
FD8E- 1090 PRBYTE .EQ $FDDA
FD8E- 1100 COUT .EQ $FDED
1110 *-----
1120 T
0800- A9 2C 1130 LDA #$1000+300
0802- 85 00 1140 STA P
0804- A9 11 1150 LDA /$1000+300
0806- 85 01 1160 STA P+1
1170 *-----
0808- A0 01 1180 .1 LDY #1
080A- B1 00 1190 LDA (P),Y
080C- F0 1C 1200 BEQ .2 TEXT LINE
080E- C9 FF 1210 CMP #$FF
0810- F0 3D 1220 BEQ .5 END OF FILE
0812- C9 D0 1230 CMP #$D0
0814- F0 09 1240 BEQ .15
0816- 20 50 08 1250 JSR PRB
0819- 88 1260 DEY
081A- B1 00 1270 LDA (P),Y
081C- 20 50 08 1280 JSR PRB
081F- 20 8E FD 1290 .15 JSR CROUT
0822- A0 01 1300 LDY #1
0824- 20 5F 08 1310 JSR BUMP
0827- 4C 08 08 1320 JMP .1
1330 *-----
082A- A0 03 1340 .2 LDY #3
082C- B1 00 1350 LDA (P),Y LENGTH OF TEXT BYTE
1360 .DO WRAP
1370 PHA SAVE FOR CR FLAG
1380 .FIN
082E- 29 7F 1390 AND #$7F
0830- AA 1400 TAX # CHARS IN LINE
0831- C8 1410 .3 INY
0832- B1 00 1420 LDA (P),Y
0834- C9 20 1430 CMP #$20 CONTROL CHAR?
0836- B0 06 1440 BCS .35 ... NO
0838- C9 0B 1450 CMP #$0B STICKY SPACE?
083A- 90 07 1460 BCC .36 ... NO
083C- A9 20 1470 LDA #$20 ...YES, JUST PRINT A SPACE
083E- 09 80 1480 .35 ORA #$80
0840- 20 ED FD 1490 JSR COUT
0843- CA 1500 .36 DEX
0844- D0 EB 1510 BNE .3
1520 .DO WRAP
1530 PLA
1540 BPL .4
1550 .FIN
0846- 20 8E FD 1560 JSR CROUT
0849- 20 5F 08 1570 .4 JSR BUMP
084C- 4C 08 08 1580 JMP .1
1590 *-----
084F- 60 1600 .5 RTS
1610 *-----
0850- 48 1620 PRB PHA
0851- A9 AE 1630 LDA #". "
0853- 20 ED FD 1640 JSR COUT
0856- A9 A4 1650 LDA #""
0858- 20 ED FD 1660 JSR COUT
085B- 68 1670 PLA
085C- 4C DA FD 1680 JMP PRBYTE
1690 *-----
085F- 98 1700 BUMP TYA
0860- 38 1710 SEC
0861- 65 00 1720 ADC P
0863- 85 00 1730 STA P
0865- 90 02 1740 BCC .1
0867- E6 01 1750 INC P+1
0869- 60 1760 .1 RTS
1770 *-----

```

Modify S-C Macro For Overlay Assembly.....Bob Sander-Cederlof

From time to time we have been asked about adding some features to the S-C Macro Assembler that would make it easy to assemble overlay segments. We have finally done it.

Some of you are developing programs that are so large that there is not room in the main 64K memory for everything at once. I have heard of one system that has over 200K bytes of machine language object code! Overlaying code segments is one method of making such a large program fit. A main program always resides in memory, and loads different sections of code when they are needed. These might be read from disk, from RAMdisk, or just copied from another section of extended memory. The main program calls program sections in one overlay, loads in another overlay and calls routines in it, and so on. The main program section also contains common subroutines that are used by more than one overlay. This is a rather standard technique on mainframe computers.

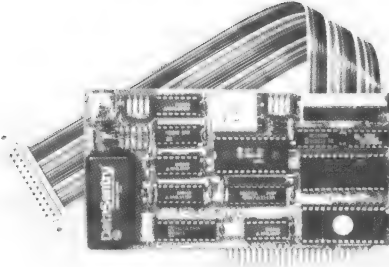
The normal method for building overlay programs on larger computers involves assembling all the parts as separate relocatable object modules, and then building the overlay "tree" of files with a program called a "Link Editor". You can do something like this with the Apple Programmer's Workshop on the IIgs. However, if you are trying to use the S-C Macro Assembler (and we hope you are!) there is no Link Editor and no way of generating relocatable object modules. The S-C Assembler does all its linking during assembly, and does not make it easy to build overlays. Of course, we are about to change that within this very article.

Since code in the main section calls subroutines in the overlays, we need a way of knowing addresses within the overlays. Likewise the overlays need to know the addresses of programs and data within the main section. If everything is assembled at the same time, all this cross linking can be done automatically. However, it requires making sure that there are no duplications of label names between overlays. The result can be huge assembly-time symbol tables that overflow the space available. By using the .INB directive (available in the ProDOS version of the S-C Macro Assembler) you can maximize the space available for symbols, but it still can run out of space.

If you are using Apple's APW on the IIgs you can do it because of the Link Editor. In fact there are Tools included in the System Loader for handling overlays. Of course you are probably trying to write programs that will run on //e's or //c's, since that is still what most people own. And you are most likely not comfortable with APW and the System Loader and all the other IIgs tools yet. So what can you do?

It is already possible to do it with the S-C Assembler, except that with such large programs the symbol table during assembly can get so large that you run out of memory. With the patching program I am about to describe you can use our assembler to build overlays, even with extremely large amounts of code that would normally overflow the space available for the symbol

Serial Pro® **The intelligent** **multifunction card from** **Applied Engineering.**



Serial Pro is a powerful multifunction card for the Apple IIe, II+, IIgs and compatible computers. Serial Pro combines a powerful serial port for interfacing to a printer, modem or other serial device, with a full function clock/calendar. Serial Pro's serial port is compatible with virtually all letter quality and dot matrix printers, including Apple's Imagewriter and Imagewriter II, and can communicate with the fastest external modems at baud rates from 50 to 19,200.

Set up is easy, because unlike other cards, there are no switches to program serial parameters or disks to use. Serial Pro has a built-in setup screen that allows all parameters to be set. And on-screen help that makes it super-easy. An on-board battery stores your selections for over 20 years, but you can change them at any time. Serial Pro is the perfect upgrade for Super Serial card owners looking for more performance or a built-in clock.

Serial Port Features

- Fully Super Serial Card compatible but more versatile and easier to use
- Compatible with printers
- Compatible with modems
- Prints graphics to printer (without software) in normal mode, 90° rotation, double size, side by side, and black/white inversion
- Variable word length, number of stop bits and parity selection
- Crystal controlled circuits are used to provide ultra accurate baud rates

Clock Calendar Features

- ProDos, DOS and AppleWorks compatible. Displays time and date on the AppleWorks screen
- Automatically time and date stamps AppleWorks files
- 20 year auto recharging nickel-cadmium battery
- Realtime interrupts
- Built-in on screen time and date settings (no software needed)
- Does not use phantom slots

Serial Pro is perfect for Apple IIe, II+, IIgs or compatible owners in need of a serial port or clock/calendar or both. But unlike other multifunction cards on the market, Serial Pro does not use phantom slots, so all the slots in your computer are useable.

Serial Pro comes complete with manual, cable and instructions for connecting to all the leading printers and modems.
 Serial Pro

\$159

Order Today! See your dealer or call Applied Engineering, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 5% sales tax. Add \$10.00 outside U.S.A.

AE APPLIED ENGINEERING™

The Apple enhancement experts

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

table during assembly. However, the following patches do not provide a COMPLETE overlaying capability. For one thing, you will have to handle loading each overlay yourself when your program executes. For another, there is not capability for using symbolic reference from the main section into the overlay sections. For another, I have only shown here patches for the ProDOS version of the S-C Macro Assembler. A similar program would work with the DOS version, but the addresses for patching would be different.

I do provide the ability for symbolic reference from overlays back to the main, which is the more important direction. For entry points within the overlay sections, you will need to use some sort of jump tables or other fixed address entry point method. Here is an outline of what I am talking about:

```

Main:  load overlay 1
       call routines in overlay 1 thru JMP table
       load overlay 2
       call routines in overlay 2 thru JMP table
       et cetera
       end

Common Subroutine A
Common Subroutine B
...
Common Subroutine Z

Common Data

Overlay n:  Jump Table:  JMP ENTRY1
                           JMP ENTRY2
                           ---
                           JMP ENTRYN

ENTRY1:  do stuff, call main routines as
         necessary, end
ENTRY2:  et cetera

```

One way of assembling such a program now without any modifications to the S-C Assembler would be to simply use .IN or .INB to include the main section withing every overlay source file. When you are assembling such a program the main section would be assembled over and over again, needlessly. The new approach lets the main section be assembled once; its object code is written on a file and its symbol table is saved in memory; then each overlay is assembled, beginning with the symbol table of the main program.

The general approach is to assemble the main section; mark the symbol table; assemble the first overlay; prune the symbol table back to the mark; assemble the next overlay; prune, assemble, and so on till the last overlay is assembled. To make it as automatic as possible, the entire assembly is controlled by an EXEC file. The EXEC file loads and executes a file named B.USR.ASM which creates the new overlay assembly capability, loads the main source code or the assembly control file for the main section, assembles the main, and then

successive assembles the various overlays.

The key, of course, is B.USR.ASM. This program, when loaded and executed, stores an address into the vector for the USR command into the S-C Macro Assembler. It also searches memory for a particular entry point inside the code which process the "ASM" command, and patches itself with that entry point. The USR command is then enabled.

The USR command works in two ways, depending on whether it is the first use since loading and executing B.USR.ASM, or a subsequent use. The first time it saves the current address for the end of the assembler's symbol table. On subsequent uses it prunes back the assembler's symbol table to that saved address. In both cases, after diddling with the symbol table, it jumps into the ASM command just after the point at which the symbol table would normally be cleared.

Perhaps it is best now to look at the listing of my program. The first section, called INSTALL.USR, is the code that is executed if you "BRUN B.USR.ASM" or "-B.USR.ASM". Lines 1180-1340 search for the spot inside the ASM command code which we need to JMP to. The code at the beginning of the ASM command looks like this:

```
STX $60
STX $EE
STX $EF
STX $53
STX $54
JSR INIT.SYM.TABLE
JSR ...
```

The place where we want to enter the ASM command is that second JSR. We still have to STX in those five locations, and the value in X needs to be zero. You can see the USR version of this at lines 2250-2340. My search loop uses the five STX lines here as a search key. When it finds the same ten bytes inside the assembler somewhere between \$9000 and \$92FF, it assumes it has found the ASM command code. By adding \$000D it computes the address of that second JSR, and then plugs that address into the JMP instruction at line 2340. Lines 1350-1420 do that computation and plugging. Lines 1430-1480 then install the USR address into the USR vector and return to the assembler.

When you type or EXEC the USR command, the assembler will jump to USR at line 1500. Lines 1510-1520 check a flag to see if this is the first time you did it or not. If it is the first time, lines 1540-1600 save the address of the end of the symbol table, and then line 1610 starts up the ASM command without clearing out the symbol table. This is how you would assemble the first overlay, after assembling the main section.

On subsequent uses of USR, lines 1630-2230 will prune back the symbol table to the same state it was the first time, and start up the assembly with that symbol table. This "pruning back" is not easy. The structure of the symbol tables is a set of 28

Multiply memory — Add up to 1 MEG with Z-RAM Ultra.

Multiply speed — Run and load programs up to 30 times faster.

Multiply production — Printer buffer lets you keep working while the system prints AppleWorks files.

Multiply value — Get Pinpoint's AppleWorks accessories with RAM enhancement software, a \$118 value, for only \$79 if ordered with Z-RAM Ultra.

Multiply efficiency — Optional 650816 16 bit processor plugs in easily with no other component changes.

Multiply applications — Run CP/M programs with built-in Z-8018 microprocessor.

IIc multiplied.

Imagine multiplying your AppleWorks word processor capacity over threefold to 22,600 lines from its present 7,250.

Multiplying your maximum number of records to 22,600 from the current 6,350. And available desktop memory rockets to over 700K with *all* of AppleWorks loaded into RAM.

You can do all of that and more with the Z-RAM Ultra expansion board for Apple IIc's.

Z-RAM Ultra completely loads programs into RAM, then runs them up to 30 times faster. And the included RAM disk is compatible with Applesoft, PRO-DOS, DOS 3.3, PASCAL and CP/M[®].

Z-RAM Ultra even expands the clipboard and autosegments large files so they can be saved on two or more disks. Nothing else even comes close to enhancing AppleWorks like this!

There's lots more.

Memory and speed just begin the story. Z-RAM Ultra 3 has a built-in Z-80B microprocessor so it can run CP/M programs. That's the largest body of software in existence and includes WordStar, dBase II, Turbo PASCAL, Microsoft BASIC and more.

With Ultra 2 & 3, there's a PRO-DOS compatible, battery-backed up clock that displays time and date on AppleWorks screens *and* time and date stamps any PRO-DOS file.

There's an AppleWorks printer buffer so you keep working while files are printed.

With its patented technology and computer aided design, Z-RAM Ultra runs with less than half the power drain of other memory cards.

Easy to install.

It's easy to get all that performance. Z-RAM Ultra installs in just a few minutes with a screwdriver — no jumper wires, no clips, no drive removal.

A complete package.

Z-RAM Ultra comes with simple, easy-to-follow instructions, RAM disk software, Z-80 operating system, CP/M manual, a five-year "hassle-free" warranty, and all the AppleWorks enhancements we're famous for.

See your dealer or call today.

For more information or to order your Z-RAM Ultra, see your dealer or call 214-241-6060 between 9 am and 11 pm, 7 days. Or send check or money order to Applied Engineering.

Prices	
Z-RAM Ultra 3 (with memory, clock and CP/M)	
265K — \$329.00	512K — \$379.00
1 MEG — \$459.00	
Z-RAM Ultra 2 (with memory and clock)	
256K — \$269.00	512K — \$319.00
1 MEG — \$399.00	
Z-RAM Ultra 1 (Memory only — expandable to 512K)	
256K — \$199.00	512K — \$249.00
Pinpoint with RAM enhancement software \$79.00	
(List price, \$118.00)	
16-bit 65C816 Upgrade \$79.00	
Z-80C Card to run CP/M software (no memory upgrade) \$159.00	
MasterCard, Visa and C.O.D. welcome. Texas residents add 6 1/4% sales tax. Add \$10.00 outside U.S.A.	

CP/M is a registered trademark of Digital Research, Inc.

AE Applied Engineering
The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006
(214) 241-6060.

linked chains. There is one chain for each letter of the alphabet, and a symbol is placed in a chain based on the first letter of its name. There are two more chains, one used for macro definitions and another for target file information. Within each chain, symbol entries are kept in alphabetical order. This means that when we assemble the first overlay, some new symbols will be linked into existing chains in arbitrary positions. The "pruning back" operations cannot simply store the original "end of symbol table" address; it must also remove any and all symbol entries from each chain which are beyond that address.

Lines 1630-2140 do just that. There are 28 pointers kept at \$0130-\$0167 which point to the first entry in each of the 28 chains. Then the first two bytes of each symbol entry contain a pointer to the next entry in alphabetical order. The last symbol entry in a chain contains 0000 in these two bytes. If there are no symbols in a particular chain, its pointer in the \$0130 area will be 0000. This code loops from 0 to 27, eliminating all entries beyond the marked end of table address for each of the 28 chains.

Finally, lines 2190-2230 restore the saved end of table address, and we fall into the code to enter the assembler without clearing the symbol table.

Using my program with an EXEC file makes all this easy. Here is an example of such an EXEC file:

```
-B.USR.ASM
LOAD MAIN
ASM
LOAD OVERLAY.1
USR
LOAD OVERLAY.2
USR
LOAD OVERLAY.3
USR
```

Each of the files loaded above could either be complete sections, or merely small assembly control files with a lot of .IN or .INB directives specifying source files to include. I recommend the latter, using the .INB directive. For example, MAIN might look like this:

```
1000      .TI 55,TITLE OF MAIN PROGRAM
1010      .INB MAIN.EQUATES
1020      .INB MAIN.PART1
1030      .INB MAIN.PART2
1040      .INB MAIN.LAST.PART
```

An OVERLAY.n file would be similar, but would name the particular files involved in that overlay. Since each overlay is separately assembled, there may be identical symbol names in different overlays with no conflict. Since the MAIN symbol table is present during the assembly of each overlay, the code in the overlays may freely reference any symbol names in the MAIN section.

```

1000 *SAVE S.USR.ASM
1010 *-----
1020 .OR $300
0300- 1030 .TF B.USR.ASM
1040 *-----
53- 1050 PAGE.NUM .EQ $53,54 Page number, starts at 0000
60- 1060 PASS.NUM .EQ $60 Pass number, starts at 00
CC- 1070 EOT .EQ $CC,CD Address of End of Symbol Table
EE- 1080 ERROR.CNT .EQ $EE,EF Error count, starts at 0000
1090 *-----
0130- 1100 HASH.TBL .EQ $0130 ... $0167 (28 pointers)
8006- 1110 USER.HOOK .EQ $8006 Address USR command jumps to
1120 *-----
53- 1130 CHAIN .EQ $53,54 Use PAGE.NUM temporarily
EE- 1140 FWD .EQ $EE,EF Use ERROR.CNT temporarily
EE- 1150 PASM .EQ $EE,EF Use ERROR.CNT temporarily
1160 *-----
1170 INSTALL.USR
1180 LDA #$9000 FIND ENTRY INTO "ASM" COMMAND
0302- 85 EE 1190 STA PASM
0304- A9 90 1200 LDA /$9000
0306- 85 EF 1210 STA PASM+1
0308- A2 03 1220 LDX #3
030A- A0 09 1230 LDY #9
030C- B1 EE 1240 .1 LDA (PASM),Y
030E- D9 BC 03 1250 .2 CMP KEY,Y
0311- F0 0A 1260 BEQ .3
0313- E6 EE 1270 INC PASM
0315- D0 F3 1280 BNE .1
0317- E6 EF 1290 INC PASM+1
0319- CA 1300 DEX
031A- D0 EE 1310 BNE .1
031C- 00 1320 BRK TRAP IF KEY NOT FOUND
031D- 88 1330 .3 DEY
031E- 10 EC 1340 BPL .2
1350 *---Install ASM link address---
0320- 18 1360 CLC
0321- A5 EE 1370 LDA PASM
0323- 69 0D 1380 ADC #$0D
0325- 8D C7 03 1390 STA JMPASM+1
0328- A5 EF 1400 LDA PASM+1
032A- 69 00 1410 ADC #0
032C- 8D C8 03 1420 STA JMPASM+2
1430 *---Install USR now-----
032F- A9 3A 1440 LDA #USR
0331- 8D 07 80 1450 STA USER.HOOK+1
0334- A9 03 1460 LDA /USR
0336- 8D 08 80 1470 STA USER.HOOK+2
0339- 60 1480 RTS
1490 *-----
033A- 2C C9 03 1500 USR
033D- 30 11 1510 BIT FIRST.TIME.FLAG
1520 BMI SUBSEQUENT.TIME
1530 *-----
1540 FIRST.TIME
033F- 38 1550 SEC Set Flag Negative for Subsequent
0340- 6E C9 03 1560 ROR FIRST.TIME.FLAG
0343- A5 CC 1570 LDA EOT Save Address of End of
0345- 8D CA 03 1580 STA PNTR SYMBOL TABLE
0348- A5 CD 1590 LDA EOT+1
034A- 8D CB 03 1600 STA PNTR+1
034D- 4C BA 03 1610 JMP JUMP.INTO.ASSEMBLER
1620 *-----
1630 SUBSEQUENT.TIME
1640 *** Rebuild pointers by starting at beginning of
1650 *** each chain in HASH TABLE and traveling to end
1660 *** of each chain, lopping off any symbols beyond
1670 *** (PNTR).
0350- A2 00 1680 LDX #0 FOR X = 0 TO 27
1690 *-----
0352- 8A 1700 .1 TXA
0353- 0A 1710 ASL INDEX * 2, CLC
0354- 69 30 1720 ADC #HASH.TBL
0356- 85 53 1730 STA CHAIN
0358- A9 01 1740 LDA /HASH.TBL
035A- 85 54 1750 STA CHAIN+1
1760 *-----

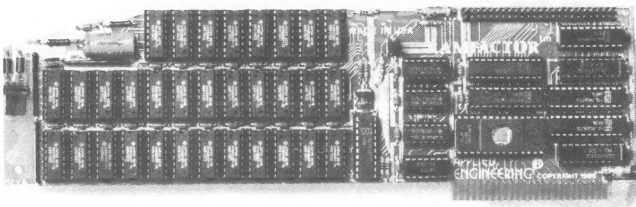
```

RamFactor™

Boot any II+, IIe, or IIGS and be up and running your favorite software in less than 1 second.

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIGS preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and an expansion connector can address an additional 16 MEG of memory for possible future expansion. Since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option, you can switch on your Apple and your favorite software will be up and running in less than 1 second!



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDos, DOS 3.3, Pascal 3.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDos clock. RamFactor will automatically segment large files so they can be saved on 5¼", 3½", and hard disks. All this performance is available on the Apple II+,

IIe, Franklin, or Laser 128 when used with an 80 column display. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

The "Instant On" Apple.



With the RamCharger battery backup option, RamFactor will retain programs and data during power failures or even when you turn your computer off. And because RamFactor plugs into slots 1-7, it's the only memory card that can cold boot an Apple without using slow floppy or hard disks. An available "Y" cable allows one RamCharger to power two RamFactors.

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDos in under a second. And with our battery back-up option, you can have permanent storage for up to 20 years.

Features:

- 256K to 1 MEG on main board. Expansion connector for possible further expansion to 16 MEG.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIGS, Franklin and Laser 128
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDos, DOS 3.3, Pascal, and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or 2.0
- Allows Apple II+ and IIe to run your AppleWorks without buying any additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDos clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	\$239
RamFactor with 512K	\$279
RamFactor with 1 MEG	\$359
RamFactor with 2-16 MEG	CALL
RamCharger backup option	\$179
"Y" cable	\$24
<small>(Allows one RamCharger to power two RamFactors.)</small>	

Order RamFactor today... with 15 day money back guarantee and our "no hassle" five year warranty. See your dealer or call (214) 241-6060, 9 a.m. to 11 p.m., 7 days, or send check or money order to Applied Engineering, MasterCard, Visa and C.O.D. welcome. Texas residents add 6% sales tax. Add \$10.00 if outside U.S.A.

AE Applied Engineering™
The Apple enhancement experts
(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

```

035C- A0 00 1770 .2 LDY #0 Get address of next symbol
035E- B1 53 1780 LDA (CHAIN),Y in this chain
0360- 85 EE 1790 STA FWD
0362- CD CA 03 1800 CMP PNTR See if beyond truncation point
0365- C8 1810 INY
0366- B1 53 1820 LDA (CHAIN),Y
0368- FO 41 1830 BEQ .5 ...end of this chain
036A- 85 EF 1840 STA FWD+1
036C- ED CB 03 1850 SBC PNTR+1
036F- B0 0A 1860 BCS .3 ...beyond truncation point
0371- A5 EE 1870 LDA FWD Step forward to next link
0373- 85 53 1880 STA CHAIN
0375- A5 EF 1890 LDA FWD+1
0377- 85 54 1900 STA CHAIN+1
0379- 90 E1 1910 BCC .2 ...ALWAYS
1920 *---Found link beyond truncation point---
037B- A0 00 1930 .3 LDY #0 Look forward through chain
037D- B1 EE 1940 LDA (FWD),Y until find one below
037F- 8D CC 03 1950 STA TCHAIN the truncation point.
0382- CD CA 03 1960 CMP PNTR
0385- C8 1970 INY
0386- B1 EE 1980 LDA (FWD),Y
0388- 8D CD 03 1990 STA TCHAIN+1
038B- ED CB 03 2000 SBC PNTR+1
038E- 90 0C 2010 BCC .4 ...below truncation point
0390- AD CC 03 2020 LDA TCHAIN Step forward to next link
0393- 85 EE 2030 STA FWD in this chain
0395- AD CD 03 2040 LDA TCHAIN+1
0398- 85 EF 2050 STA FWD+1
039A- B0 DF 2060 BCS .3 ...ALWAYS
2070 *---Patch out truncated links of chain---
039C- A0 00 2080 .4 LDY #0
039E- AD CC 03 2090 LDA TCHAIN
03A1- 91 53 2100 STA (CHAIN),Y
03A3- C8 2110 INY
03A4- AD CD 03 2120 LDA TCHAIN+1
03A7- 91 53 2130 STA (CHAIN),Y
03A9- 90 B1 2140 BCC .2 ...ALWAYS
2150 *-----
03AB- E8 2160 .5 INX NEXT X (0...27)
03AC- E0 1C 2170 CPX #28
03AE- 90 A2 2180 BCC .1
2190 *---Put EOT back too-----
03B0- AD CA 03 2200 LDA PNTR
03B3- 85 CC 2210 STA EOT
03B5- AD CB 03 2220 LDA PNTR+1
03B8- 85 CD 2230 STA EOT+1
2240 *-----
2250 JUMP.INTO.ASSEMBLER
03BA- A2 00 2260 LDX #0
2270 KEY
03BC- 86 60 2280 STX PASS.NUM
03BE- 86 EE 2290 STX ERROR.CNT
03C0- 86 EF 2300 STX ERROR.CNT+1
03C2- 86 53 2310 STX PAGE.NUM
03C4- 86 54 2320 STX PAGE.NUM+1
2330 JMPASM
03C6- 4C 00 00 2340 JMP *-# *** ADDRESS FILLED IN BY PROGRAM ***
2350 *-----
03C9- 00 2360 FIRST.TIME.FLAG .HS 00
03CA- 00 00 2370 PNTR .HS 00.00
03CC- 00 00 2380 TCHAIN .HS 00.00
2390 *-----

```

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is \$18 per year in the USA, sent Bulk Mail; add \$3 for First Class postage in USA, Canada, and Mexico; add \$14 postage for other countries. Back issues are \$1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is \$64 per year in the US, Canada and Mexico, and \$87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)